Architecture Evaluation Methods

Presenter: Alexandru Chica

Contents

Architecture

- What is an architecture?
- Quality attributes of an architecture

Evaluating an architecture

- Why evaluate an architecture?
- Benefits and costs
- Different approaches:
  - SAAM (Software Architecture Analysis Method)
  - ATAM (Architecture Tradeoff Analysis Method) (tbd.)
  - ARID (Active reviews for intermediate designs) (tbd.)

Architecture

What is an architecture?

- The software architecture of a program or computing system is the structure or structures of the system, which comprises of software components, the externally visible properties of those components, and the relationships among them. [Bass 98]

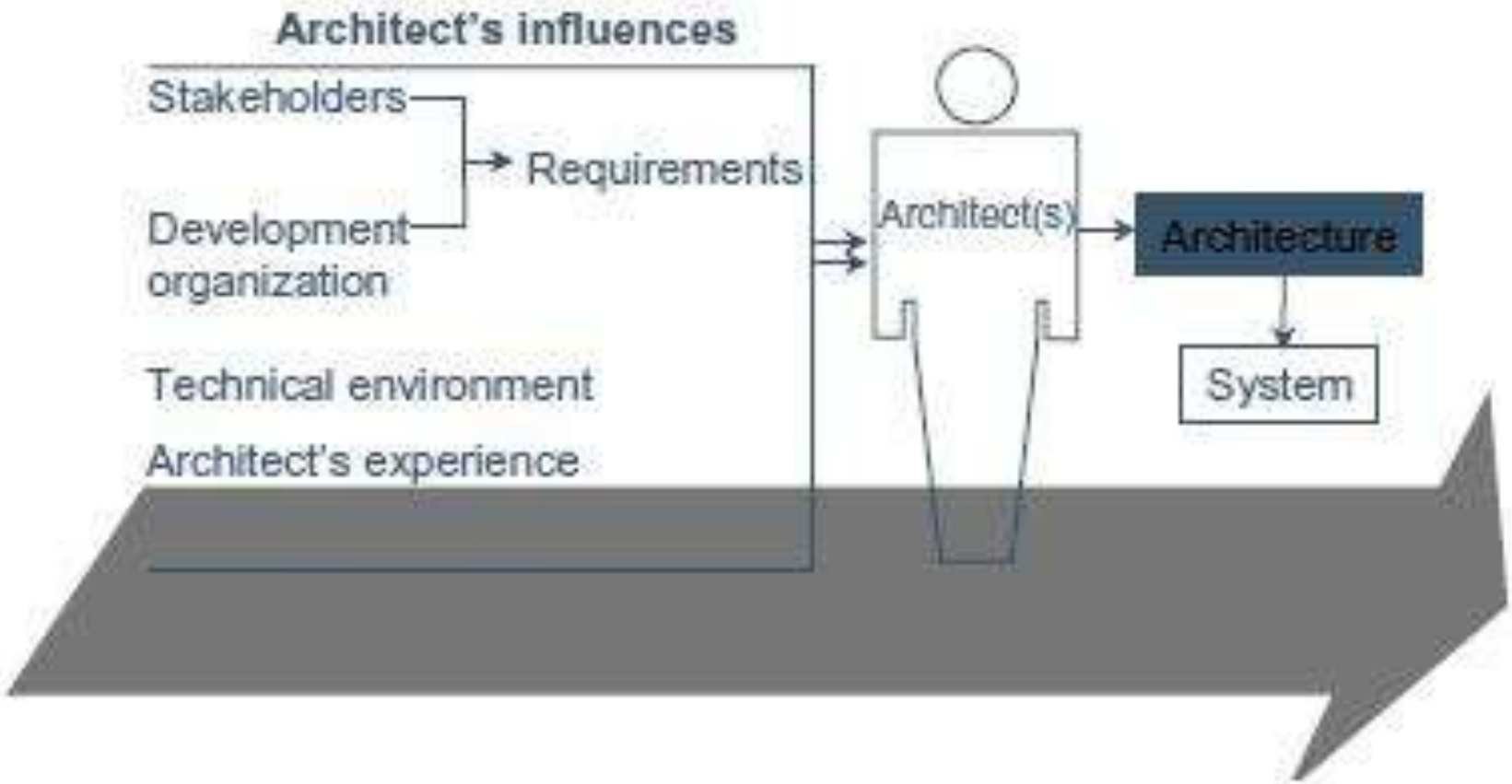
- Architecture is high-level design

- Architecture is the overall structure of a system

- ***Architecture is components and connectors***

What is an architecture?

Architecture



Architecture

Quality attributes of an architecture (I)

- **Usability** - the measure of a user's ability to utilize a system effectively

- **Functionality** - the ability of the system to do the work for which it was intended

- **Modifiability** - the ability to make changes to a system quickly and cost effectively

- **Subsetability**

- **Reliability** - the ability of the system to keep operating over time

Architecture

Quality attributes of an architecture (II)

- **Conceptual integrity –** the architecture should do similar things in similar ways

- **Performance -** the responsiveness of the system

- **Availability –** the proportion of time the system is up and running (the delay between failures and time needed to resume normal operations)

- **Testability**

- **Security**

Evaluating an architecture

Why evaluate an architecture?

- The earlier you find a problem in a SW project, the better off you are (the cost to fix an error in early design phase is much smaller than the cost to fix the same error in implementation/testing)

- Architecture is the earliest point in the project where trade-offs are visible

- Architecture determines the structure of the project: schedules, budgets, performance indicators, team structure, testing and maintenance activities

- Risk management

Evaluating an architecture

Benefits and costs

- (+) Forces clear explanation of architecture
- (+) Puts stakeholders in the same room
- (+) Identifies and solves conflicting goals
- (+) Forces clarification of specific quality goals
- (+) Identifies risks early in the lifecycle
- (-) Costs time and money

Any kind of organized approach to evaluation is way better than none

Evaluating an architecture

SAAM (Software Architecture Analysis Method)

- Based on scenarios
  - A scenario represents a description of a stakeholder's interaction with the system
- Scenarios are created depending on the point of view of each stakeholder:
  - **Developer –** interested in reusability, implementation, maintenance
  - **Project Manager –** interested in time, cost, quality, extensibility
  - **Tester –** interested in usability, mapping to requirements

Evaluating an architecture

Steps of a SAAM evaluation

Identify and assemble stakeholders

↓

Develop and prioritize scenarios

↓

Describe architecture (actual review)

↓

Classify scenarios as direct or indirect

↓

Perform scenario evaluation

↓

Reveal scenario interactions

↓

Generate overall evaluation

Evaluating an architecture

SAAM scenarios

- Scenarios should refer to the evolution that the system must support (based on requirements)
  - Functionality
  - Development activities
  - Change activities
- Scenarios should represent tasks relevant to all stakeholders
- Suggestion: 10-15 prioritized scenarios
- Scenarios can be classified in two classes
  - Direct scenarios do not require system modifications
  - Indirect scenarios require system change

Evaluating an architecture

SAAM scenario evaluation

- For each direct scenario, see if scenario can be performed with current system state
- For each indirect scenario
  - Identify the components which have to be modified, added or deleted
  - Estimate the difficulty of the modification (based on the number of components to be modified and the effect of the modification)

Evaluating an architecture

SAAM scenario interaction

- Multiple indirect scenarios affecting the same component could indicate a problem
  - Could be good: if scenarios are variants of each other
  - Could be bad: indicates a poor separation of responsibilities

SAAM Evaluation Results

- Classification of scenarios based on importance
- Decision if architecture is accepted or has to be modified

Evaluating an architecture

Examples

- Indirect scenarios:
  - Extension of capabilities – adding new functionality, enhancing existing functionality
  - Deletion of unwanted capabilities
  - Adaption to new operating environments (hardware, OS, I/O devices)
  - Restructuring – modularizing, optimizing, creating reusable components

Evaluating an architecture

Examples

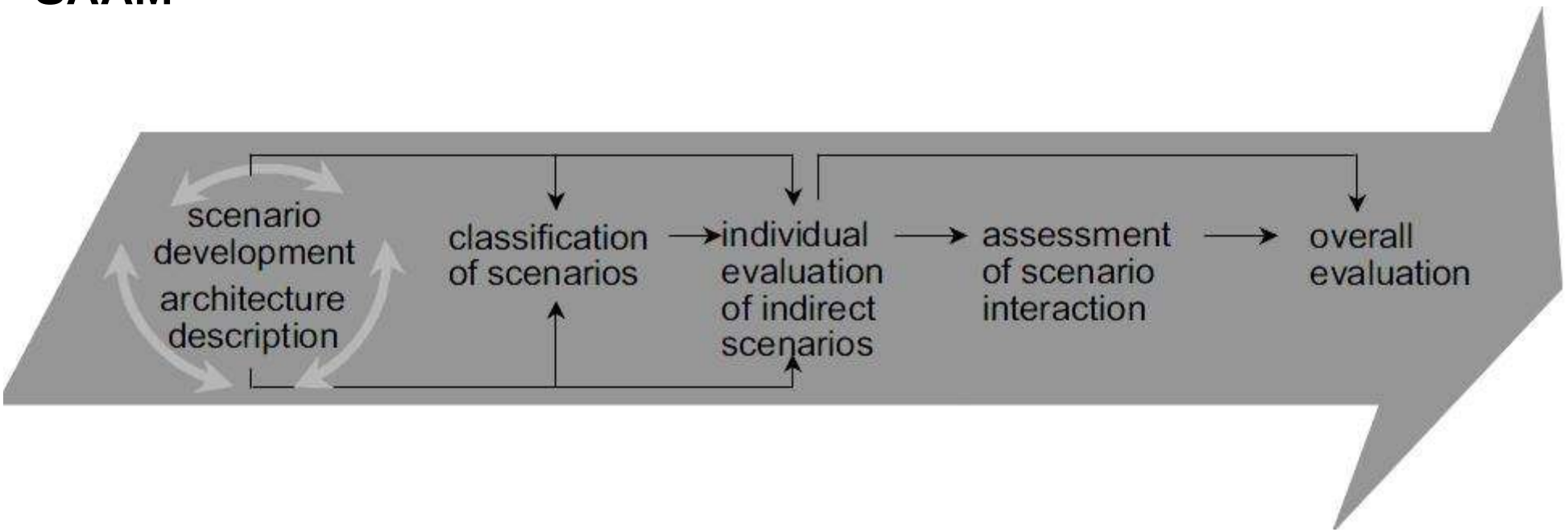
- Direct scenarios
  - Confronting the architecture with regular use cases
  - Use logic that is provided by the interfaces
  - Stress testing – behavior of components in case of intensive usage
  - Corruption of data/components after long-term usage
  - Data integrity when sending it through communication channels
  - Scenarios regarding functionality found in the requirements
  - Ease of test – how easy is it to test a requirement

Evaluating an architecture

Conclusion

Any kind of organized approach to evaluation is way better than none.

SAAM



Evaluating an architecture

?